Journal of Computational and Applied Mathematics 234 (2010) 1442–1447

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Application of the variational iteration method for solving nth-order integro-differential equations

Xufeng Shang [a,*], Danfu Han [b]

[a] Department of Information and Mathematics Sciences, China Jiliang University, 310018, Hangzhou, China

[b] Department of Mathematics, Zhejiang University, 310027, Hangzhou, China

ARTICLE INFO

Article history:
Received 14 December 2008
Received in revised form 1 February 2010

MSC:
65Rxx

Keywords:
Variational iteration method
Integro-differential equations
The system of ordinary integro-differential equations

ABSTRACT

In this paper, we solve nth-order integro-differential equations by changing the problem to a system of ordinary integro-differential equations and using the variational iteration method. The variational iteration formula is derived and the Lagrange multiplier can be effectively identified. It is well known that one of the advantages of He's variational iteration method is the free choice of initial approximation. Therefore, we use this advantage to construct an initial values without unknown parameters. Some examples are given and the results reveal that the method is very effective and simple compared with the Homotopy perturbation method (HPM).

1. Introduction

In this paper, we extend the applications of the variational iteration method (VIM) to find approximate solutions for the general nth-order integro-differential equations. The variational iteration method which proposed in [1,2], is effectively and easily used to solve some classes of nonlinear problems. For linear problems, its exact solution can be obtained by only one iteration step due to the fact that the Lagrange multiplier can be exactly identified. Recently, the application of the VIM in linear and nonlinear problems has been devoted by scientists and engineers, for example, nonlinear systems of ordinary differential equations [3], integral equations [4], boundary-value problems [5], integro-differential equations [6,11].

We consider the general nth-order integro-differential equations of the type [7]:

$$y^{(n)}(x) + f(x)y(x) + \int_a^b k(x,t)y^{(m)}(t)\mathrm{d}t = g(x), \quad a < x < b \tag{1}$$

with initial conditions

$$y(a) = \alpha_0,\ y'(a) = \alpha_1,\ y''(a) = \alpha_2, \ldots,\ y^{(n-1)}(a) = \alpha_{n-1},$$

where $\alpha_i, i = 0, 1, \ldots, n-1$, are real constants, m and n are integers and $m < n$. In Eq. (1) the functions f, g and k are given, and y is the solution to be determined. We assume that Eq. (1) has the unique solution. In this paper, we change the problem to a system of ordinary integro-differential equations and apply the variational iteration method to solve it, so that the Lagrange multiplier can be effectively identified.

* Corresponding author.
E-mail address: xfshang32@yahoo.com.cn (X. Shang).

0377-0427/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2010.02.020

2. Variational iteration method

Using the transformation

$$y = y_1,\ \frac{dy}{dx} = y_2,\ \frac{d^2y}{dx^2} = y_3, \dots,\ \frac{d^{(n-1)}y}{dx^{(n-1)}} = y_n,$$

we can rewrite the integro-differential equation (1) as the system of ordinary integro-differential equations:

$$\begin{cases} \dfrac{dy_1}{dx} = y_2, \\ \dfrac{dy_2}{dx} = y_3, \\ \dfrac{dy_3}{dx} = y_4, \\ \vdots \\ \dfrac{dy_n}{dx} = g(x) - f(x)y_1(x) - \displaystyle\int_a^b k(x,t)y_{m+1}(t)dt \end{cases} \tag{2}$$

with initial conditions

$$y_1(a) = \alpha_0,\ y_2(a) = \alpha_1,\ y_3(a) = \alpha_2, \dots,\ y_n(a) = \alpha_{n-1}.$$

To illustrate the basic concepts of the VIM, we consider the following differential equation:

$$L[u(x)] + N[u(x)] = g(x),$$

where L is a linear operator, N is a nonlinear operator and $g(x)$ is a given continuous function.

The basic character of the method is to construct a correction functional for the system, which reads

$$u_{n+1}(x) = u_n(x) + \int_a^x \lambda[Lu_n(t) + N\tilde{u}_n(t) - g(t)]dt, \tag{3}$$

where λ is a general Lagrange multiplier which can be identified optimally via variational theory, u_n is the nth approximate solution, and $\tilde{u}_n$ denotes a restricted variation, i.e. $\delta\tilde{u}_n = 0$.

According to the variational iteration method, to solve the system (2), we can construct the following correction functionals:

$$y_j^{(k+1)}(x) = y_j^{(k)}(x) + \int_a^x \lambda_j(x,t)[y_j'^{(k)}(t) - \tilde{y}_{(j+1)}^{(k)}(t)]dt, \quad j = 1, 2, \dots, n-1,$$

$$y_n^{(k+1)}(x) = y_n^{(k)}(x) + \int_a^x \lambda_n(x,t)\left[y_n'^{(k)}(t) - g(t) + f(t)\tilde{y}_1^{(k)}(t) + \int_a^b k(t,s)\tilde{y}_{m+1}^{(k)}(s)ds\right]dt,$$

where the subscript (k) is the number of iteration steps.

Calculating variation with respect to $y_j^{(k)}$ $(j = 1, 2, \dots, n)$, respectively, and noting that $\delta y_j^{(k)}(0) = 0$, we have

$$\begin{aligned} \delta y_j^{(k+1)}(x) &= \delta y_j^{(k)}(x) + \delta\int_a^x \lambda_j(x,t)[y_j'^{(k)}(t) - \tilde{y}_{(j+1)}^{(k)}(t)]dt \\ &= \delta y_j^{(k)}(x) + \lambda_j(x,t)\delta y_j^{(k)}(t)|_{t=x} - \int_a^x \frac{\partial\lambda_j(x,t)}{\partial t}\delta y_j^{(k)}(t)dt \\ &= (1 + \lambda_j(x,x))\delta y_j^{(k)}(x) + \int_a^x \left(-\frac{\partial\lambda_j(x,t)}{\partial t}\right)\delta y_j^{(k)}(t)dt = 0, \quad j = 1, 2, \dots, n-1, \\ \delta y_n^{(k+1)}(x) &= \delta y_n^{(k)}(x) + \delta\int_a^x \lambda_n(x,t)\left[y_n'^{(k)}(t) - g(t) + f(t)\tilde{y}_1^{(k)}(t) + \int_a^b k(t,s)\tilde{y}_{m+1}^{(k)}(s)ds\right]dt \\ &= \delta y_n^{(k)}(x) + \lambda(x,t)\delta y_n^{(k)}(t)|_{t=x} - \int_a^x \frac{\partial\lambda_n(x,t)}{\partial t}\delta y_n^{(k)}(t)dt \\ &= (1 + \lambda_n(x,x))\delta y_n^{(k)}(x) + \int_a^x \left(-\frac{\partial\lambda_n(x,t)}{\partial t}\right)\delta y_n^{(k)}(t)dt = 0. \end{aligned}$$

For arbitrary $\delta y_j^{(k)}$, $j = 1, 2, \dots, n$, the following stationary conditions are obtained:

$$-\frac{\partial\lambda_1(x,t)}{\partial t} = -\frac{\partial\lambda_2(x,t)}{\partial t} = \cdots = -\frac{\partial\lambda_n(x,t)}{\partial t} = 0,$$

and the natural boundary condition:

$$1 + \lambda_j(x,x) = 0, \quad j = 1, 2, \dots, n.$$

The Lagrange multipliers, therefore, can be identified as

$$\lambda_j(x,t) = -1, \quad j = 1, 2, \ldots, n$$

and the following iteration formula can be obtained as

$$y_j^{(k+1)}(x) = y_j^{(k)}(x) - \int_a^x [y_j'^{(k)}(t) - y_{j+1}^{(k)}(t)]\mathrm{d}t, \quad j = 1, 2, \ldots, n-1,$$

$$y_n^{(k+1)}(x) = y_n^{(k)}(x) - \int_a^x \left[y_n'^{(k)}(t) - g(t) + f(t)y_1^{(k)}(t) + \int_a^b k(t,s)y_{m+1}^{(k)}(s)\mathrm{d}s\right]\mathrm{d}t. \quad (4)$$

Beginning with $y_1^{(0)}(x) = \alpha_0, y_2^{(0)}(x) = \alpha_1, y_3^{(0)}(x) = \alpha_2, \ldots, y_n^{(0)}(x) = \alpha_{n-1}$, by the iteration formula (4), we can obtain the numerical solution of Eq. (1).

3. Convergence

Generally one iteration leads to high accurate solution by the variational iteration method if the initial solution is carefully chosen with some unknown parameters. If we begin with $u_0(x) = u(0)$, an series solution can be obtained. The convergence of the method is systematically discussed in [8], and comparison of the method with the Adomian method was conducted by many authors via illustrative examples, especially Wazwaz gave a completely comparison between the two methods [9], revealing that the variational iteration method has many merits over the Adomian method; it can completely overcome the difficulty arising in the calculation of the Adomian polynomial. Though the variational iteration method leads to fast convergent solutions, unnecessary calculation arises in the solution procedure. The demerit of this method is illustrated completely from an example in [10].

In the following section, we give a comparison between the VIM and the HPM from some examples.

4. Application

In this section, in order to verify numerically whether the proposed method leads to higher accuracy, we evaluate the numerical solution of the problem (1). To show the efficiency of the present method for our problem in comparison with the exact solution we evaluate the absolute error defined by

$$E_k = |y(x) - y^{(k)}(x)|, \quad \text{for } k = 1, 2, \ldots,$$

where $y(x)$ is the exact solution, and $y^{(k)}(x)$ is the kth-iteration approximate solution. Moreover, we compare the CPU times of the VIM and the Homotopy Perturbation method (HPM) for the problem (1) to obtain the same solution. The computations associated with the examples were performed using Mathematica 5.2.

Example 1. Consider the second-order integro-differential equation

$$\begin{cases} y''(x) = \mathrm{e}^x - x + \int_0^1 xty(t)\mathrm{d}t \\ y(0) = 1, y'(0) = 1. \end{cases} \quad (5)$$

The exact solution for this problem is $y(x) = \mathrm{e}^x$.

We solve (5) by using the method in Section 2. Starting with initial approximations $y_1^{(0)}(x) = 1,\ y_2^{(0)}(x) = 1$, by the iteration formula (4), we can obtain the following results:

$$y_1^1(x) = 1 + x,$$

$$y_1^2(x) = \mathrm{e}^x - \frac{x^3}{12},$$

$$y_1^5(x) = \mathrm{e}^x - \frac{x^3}{1080},$$

$$y_1^{10}(x) = \mathrm{e}^x - \frac{x^3}{9720000},$$

$$y_1^{15}(x) = \mathrm{e}^x - \frac{x^3}{26244000000}.$$

Table 1 shows the numerical results of the example at $x = 0.2(0.2)1.0$. As we see from Table 2, it is clear that the CPU Times of the present method is less than that of the HPM [7].

Table 1
Numerical results for Example 1.

x	$E_k = \|y(x) - y^{(k)}(x)\|$		
	VIM, $k = 5$	VIM, $k = 10$	VIM, $k = 15$
0.2	7.4074e−06	8.2305e−10	3.0483e−13
0.4	5.9259e−05	6.5844e−09	2.4387e−12
0.6	2.0000e−04	2.2222e−08	8.2305e−12
0.8	4.7407e−04	5.2675e−08	1.9509e−11
1.0	9.2593e−04	1.0288e−07	3.8104e−11

Table 2
CPU times for Example 1.

Steps	CPU times (s)	
Method	VIM	HPM
$k = 5$	0.062	0.156
$k = 10$	0.125	0.172
$k = 15$	0.141	0.234
$k = 30$	0.422	0.491
$k = 40$	0.562	0.569
$k = 50$	0.672	0.719

Table 3
Numerical results for Example 2.

x	$E_k = \|y(x) - y^{(k)}(x)\|$		
	VIM, $k = 5$	VIM, $k = 10$	VIM, $k = 15$
0.2	2.1251e−05	6.3040e−07	6.9948e−08
0.4	3.4002e−04	1.0086e−05	1.1192e−06
0.6	1.7214e−03	5.1063e−05	5.6658e−06
0.8	5.4403e−02	1.6138e−04	1.7907e−05
1.0	1.3282e−02	3.9400e−04	4.3718e−05

Example 2. Consider the third-order integro-differential equation

$$\begin{cases} y'''(x) = \sin(x) - x - \displaystyle\int_0^{\pi/2} xty'(t)\mathrm{d}t \\ y(0) = 1, y'(0) = 0, y''(0) = -1. \end{cases} \tag{6}$$

The exact solution for this problem is $y(x) = \cos x$.

We solve (6) by using the method in Section 2. Starting with initial approximations $y_1^{(0)}(x) = 1, y_2^{(0)}(x) = 0, y_3^{(0)}(x) = -1$, by the iteration formula (4), we can obtain the following results:

$$\begin{aligned} y_1^1(x) &= 1, \\ y_1^2(x) &= 1 - \frac{x^2}{2}, \\ y_1^5(x) &= \cos(x) + \frac{\pi^5}{23040}x^4, \\ y_1^{10}(x) &= \cos(x) - \frac{\pi^{18}}{509607936000}x^4 + \frac{\pi^{15}}{21233664000}x^4, \\ y_1^{15}(x) &= \cos(x) - \frac{\pi^{30}}{1878618695270400000 0}x^4. \end{aligned}$$

Table 3 shows the numerical results of the example at $x = 0.2(0.2)1.0$. As we see from Table 4, it is clear that the CPU Times of the present method is less than that of the HPM [7].

Example 3. Consider the eight-order integro-differential equation

$$\begin{cases} y^{(8)}(x) = -8e^x + x^2 + y(x) + \displaystyle\int_0^1 x^2y'(t)\mathrm{d}t \\ y(0) = 1, y'(0) = 0, y''(0) = -1, y'''(0) = -2, y^{(4)}(0) = -3, \\ y^{(5)}(0) = -4, y^{(6)}(0) = -5, y^{(7)}(0) = -6. \end{cases} \tag{7}$$

Table 4
CPU times for Example 2.

Steps	CPU times (s)	
Method	VIM	HPM
$k=5$	0.093	0.297
$k=10$	0.250	0.421
$k=15$	0.344	0.547
$k=30$	0.985	1.203
$k=40$	1.375	1.7034
$k=50$	1.687	1.827

Table 5
Numerical results for Example 3.

x	$E_k = \lvert y(x) - y^{(k)}(x)\rvert$		
	VIM, $k=5$	VIM, $k=10$	VIM, $k=15$
0.2	4.6014e−07	2.9088e−14	1.1102e−16
0.4	3.0515e−05	3.1107e−11	1.1657e−14
0.6	3.6048e−04	1.8626e−09	6.6203e−13
0.8	2.1025e−03	3.4362e−08	1.1751e−11
1.0	8.3333e−03	3.3265e−07	1.0971e−10

Table 6
CPU times for Example 3.

Steps	CPU times (s)	
Method	VIM	HPM
$k=5$	0.141	0.593
$k=10$	0.422	1.843
$k=15$	0.859	5.515
$k=30$	4.641	46.234
$k=40$	8.109	110.813

The exact solution for this problem is $y(x) = (1-x)e^x$.

We solve (7) by using the method in Section 2. Starting with initial approximations $y_1^{(0)}(x) = 1, y_2^{(0)}(x) = 0, y_3^{(0)}(x) = -1, y_4^{(0)}(x) = -2, y_5^{(0)}(x) = -3, y_6^{(0)}(x) = -4, y_7^{(0)}(x) = -5, y_8^{(0)}(x) = -6$, by the iteration formula (4), we can obtain following results:

$$y_1^1(x) = 1,$$

$$y_1^2(x) = 1 - \frac{x^2}{2},$$

$$y_1^5(x) = 1 - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{8} - \frac{x^5}{30},$$

$$y_1^{10}(x) = 9 - 8e^x + 8x + \frac{7x^2}{2} + x^3 + \frac{5x^4}{24} + \frac{x^5}{30} + \frac{x^6}{240} + \frac{x^7}{2520} + \frac{x^8}{40320},$$

$$y_1^{15}(x) = 9 - 8e^x + 8x + \frac{7x^2}{2} + x^3 + \frac{5x^4}{24} + \frac{x^5}{30} + \frac{x^6}{240} + \frac{x^7}{2520} + \frac{x^8}{40320} - \frac{2519x^{10}}{9144576000} - \frac{x^{11}}{19958400} - \frac{x^{12}}{159667200} - \frac{x^{13}}{1556755200} - \frac{x^{14}}{17435658240} - \frac{x^{15}}{217945728000}.$$

Table 5 shows the numerical results of the example at $x = 0.2(0.2)1.0$. In fact, to obtain the same solution, our method is faster than the HPM from the CPU times in Table 6.

5. Conclusion

This paper, nth-order integro-differential equations are solved using the variational iteration method. We change the problem to a system of ordinary integro-differential equations, so that the Lagrange multiplier can be effectively identified. It is well known that one of the advantages of He's variational iteration method is the free choice of initial approximation. Therefore, in this paper, we use this advantage to construct initial values without unknown parameters. Some examples are given and the results reveal that the method is very effective. Some of the nonlinear and linear equations are examined by the modified method to illustrate the effectiveness and convenience of this method, and in all cases, the modified technique performed excellently. The results reveal that the proposed method is very effective and simple.

Acknowledgements

The authors are very grateful to one of the reviewers for his helpful suggestions.

References

[1] J.H. He, Variational iteration method-a kind of non-linear analytical technique: Some examples, Int. J. Nonlinear Mech. 34 (4) (1999) 699–708.
[2] J.H. He, Variational iteration method Some recent results and new interpretations, J. Comput. Appl. Math. 207 (2007) 3–17.
[3] J. Biazar, H. Ghazvini, He's variational iteration method for solving linear and non-linear systems of ordinary differential equations, Appl. Math. Comput. 191 (2007) 287–297.
[4] Lan Xu, Variational iteration method for solving integral equations, Comput. Math. Appl. 54 (2007) 1071–1078.
[5] M. Hesaaraki, Y. Jalilian, A numerical method for solving *n*th-order boundary-value problems, Appl. Math. Comput. 196 (2) (2008) 889–897.
[6] Sh.Q. Wang, J.H. He, Variational iteration method for solving integro-differential equations, Phys. Lett. A 367 (2007) 188–191.
[7] A. Golbabai, M. Javidi, Application of He's homotopy perturbation method for *n*th-order integro-differential equations, Appl. Math. Comput. 190 (2007) 1409–1416.
[8] M. Tatari, M. Dehghan, On the convergence of He's variational iteration method, J. Comput. Appl. Math. 207 (1) (2007) 121–128.
[9] A.M. Wazwaz, A comparison between the variational iteration method and Adomian decomposition method, J. Comput. Appl. Math. 207 (1) (2007) 129–136.
[10] J.H. He, X.H. Wu, Variational iteration method: New development and applications, Comput. Math. Appl. 54 (2007) 881–894.
[11] J. Biazar, H. Aminikhah, A new technique for solving nonlinear integral-differential equations, Comput. Math. Appl. (2009) doi:10.1016/j.camwa.2009.03.042.